

## CLAIMS

I/We claim:

- [c1]            1.     An architecture in a computer system for execution of processes using an integration environment, comprising:
- an application service interface for each application that interfaces with a process, the application service interfaces being independent of the integration environment; and
- a common service interface for each type of service provided through the application service interfaces, the common service interface for a type of service being independent of the application service interfaces through which that type of service is provided.
- [c2]            2.     The architecture of claim 1 whereby the processes interface with the common service interfaces so that the processes are independent of the particular applications selected to support the types of services and independent of the particular integration environment selected.
- [c3]            3.     The architecture of claim 1 wherein the integration environment is provided by an integration server.
- [c4]            4.     The architecture of claim 1 wherein the application service interfaces and the common service interfaces are used for an application to access a service of a process and for a process to access services of an application.
- [c5]            5.     The architecture of claim 1 wherein the processes are integration processes that are independent of the particular integration environment and particular applications that support a type of service.

[c6] 6. A computer-readable medium containing instructions for controlling an integration server to execute integration processes that access types of services, by a method comprising:

for each application that provides a type of service accessed by the integration processes, providing an implementation of an application service interface, the application service interface for accessing the application in a manner that is integration server independent and application dependent; and

for each type of service accessed by the integration processes,

for each application that provides that type of service, providing an implementation of a common service interface, the common service interface for accessing applications that provide that type of service in a manner that is integration server independent and application independent, the implementation being application dependent and using the application service interface to access the application in a manner that is integration server independent.

[c7] 7. The computer-readable medium of claim 6 wherein a function of the application service interface for an application that provides a type of service translates its invocation by a function of the common service interface for that type of service to an invocation of one or more functions of the application and translates its invocation by the application to invocations of one or more functions of the common service interface.

[c8] 8. The computer-readable medium of claim 6 wherein a function of the common service interface for a type of service transforms its invocation by integration processes to invocations of one or more functions of an application service interface for an application that provides that type of service and

transforms its invocation by a function of an application service interface of the application to invocations of one or more functions of the integration processes.

[c9]            9.     The computer-readable medium of claim 6 wherein the implementations are provided by XSLT transforms.

[c10]           10.    The computer-readable medium of claim 6 wherein the common service interface and the application service interface are defined using WSDL.

[c11]           11.    A method for executing an integration process on an integration server, the integration process being independent of integration servers and applications that provide a type of service, the method comprising:

              for each type of service accessed by the integration process,

                  providing a common service interface that is independent of the applications that provide that type of service; and

                  providing an implementation of the common service interface for an application that provides that type of service, the implementation being application and integration server dependent; and

              performing instructions of the integration process that use the common service interface to access a certain type of service of an application and that provide services to the application of that type of application that uses the common service interface.

[c12]           12.    The method of claim 11 wherein the provided implementations use an application service interface for each application that is independent of the integration servers.

[c13]           13.    The method of claim 12 including providing an implementation of the application service interface for each application.

[c14]           14.    The method of claim 13 wherein an implementation of an application service interface for an application translates invocations of functions of the application service interface by an implementation of a function of a common service interface to invocations of one or more functions of the application and translates invocations of functions of the application service interface by the application to invocations of one or more functions of the common service interface.

[c15]           15.    The method of claim 11 wherein the provided implementations transform invocations of functions of the common service interface by the integration process to invocations of one or more functions of an application and transform invocations of functions of the common service interface by an application to invocations of one or more functions of the integration process.

[c16]           16.    The method of claim 15 wherein the application invokes the common service interface through an application service interface.

[c17]           17.    The method of claim 11 wherein the common service interface is defined using WSDL.

[c18]           18.    The method of claim 11 wherein the implementations are provided by XSLT transforms.

[c19]           19.    A method in a computer system for providing an integration environment in which integration processes can be developed independent of integration servers and applications that provide types of services, the method comprising:

                providing an application service interface for each application that is independent of the integration server upon which the integration process is to execute;

providing a common service interface for each type of service that is independent of the applications that provide that type of service and that is independent of the integration server upon which the integration process is to execute; and  
for each type of service, providing instructions that  
transform invocations of functions of the common service interface for that type of service by an integration process to invocations of one or more functions of an application service interface of an application that provides that type of service, and  
translate invocations of functions of the application service interface by an application that provides that type of service to an integration process to invocations of one or more functions of the common service interface for that type of service  
so that integration processes can be developed to provide services to an application that provides that type of service via an application service interface and to use services of an application that provides that type of service independent of the application that provides that service and independent of the integration server upon which the integration process is executing.

[c20] 20. The method of claim 19 wherein the application service interface and the common service interface are defined using WSDL.

[c21] 21. The method of claim 19 wherein the provided instructions that transform invocations of functions are XSLT transforms.

[c22] 22. The method of claim 19 including providing functions for mapping identifiers from different applications for the same information to a common identifier.

[c23] 23. The method of claim 19 including providing definitions of common objects that are used by the integration processes.

[c24] 24. The method of claim 19 wherein the instructions that transform provide semantic transformation between the application service interface and the common service interface.

[c25] 25. The method of claim 19 wherein an integration server provides an interface to an application and including providing a translator that translates the interface provided by the integration server to the application service interface for that application.

[c26] 26. A method in a computer system for providing an integration environment in which integration processes can be developed independent of integration servers, the method comprising:

providing an application service interface for each application that is independent of the integration server upon which the integration process is to execute; and

for each application, providing instructions that

translate invocations of functions of the application service interface for that application by an integration process to invocations of one or more functions of an integration server interface of that application, and

translate invocations of functions of the integration server interface for that application by that application to invocations of one or more functions of the application service interface for that application that execute instructions of the integration process

so that an integration process can be developed to use the application service interface to provide services to an application and to use

services of an application independent of the integration server upon which the integration process is executing.

[c27]            27.    The method of claim 26 wherein the application service interface is defined using WSDL.

[c28]            28.    The method of claim 26 including providing functions for mapping identifiers from different applications for the same information to a common identifier.

[c29]            29.    The method of claim 26 including providing definitions of common objects that are used by the integration processes.

[c30]            30.    The method of claim 26 wherein the instructions that translate provide syntactic translation between the application service interface and the integration server interface.

[c31]            31.    The method of claim 26 wherein the integration server interface of an application provides an interface to an application program interface provided by the application.

[c32]            32.    A method in a computer system for providing an integration environment in which integration processes can be developed independent of integration servers and applications that provide a type of service, the method comprising:

                 providing a common service interface for each type of service that is independent of the application that is providing the type of service and that is independent of the integration server upon which an integration process is to execute; and  
                 for each type of service, providing instructions that

convert invocations of functions of the common service interface for that type of service by an integration process to invocations of one or more functions of the integration server interface of the application that provides that type of service; and

convert invocations of functions of the integration service interface by an application that provides that type of service to invocations of one or more functions of the common service interface for that type of service

so that an integration process can be developed to use the common service interface to provide services to an application that provides a type of service and to use that type of service independent of the application that provides that type of service and independent of the integration server upon which the integration process is executing.

[c33] 33. The method of claim 32 wherein the common service interface is defined using WSDL.

[c34] 34. The method of claim 32 wherein the provided instructions that convert invocations of functions perform a semantic transformation of function invocations between the common service interface and an application service interface and perform a syntactic translation of function invocations between the application service interface and an integration server interface.

[c35] 35. The method of claim 32 including providing functions for mapping identifiers from different applications for the same information to a common identifier.

[c36] 36. The method of claim 32 including providing definitions of common objects that are used by the integration processes.



[c37] 37. An integration server for executing integration processes that access types of services, the integration processes being independent of the integration server and independent of applications that provide each type of service, comprising:

for each application that provides a type of service accessed by the integration processes, an implementation of an application service interface, the application service interface for accessing the application in a manner that is integration server independent, the implementation being integration server dependent; and

for each type of service accessed by the integration processes,

for each application that provides that type of service, an implementation of a common service interface, the common service interface for accessing applications that provide that type of service in a manner that is application independent, the implementation being application dependent and using the application service interface to access the application in a manner that is integration server independent.

[c38] 38. The integration server of claim 37 wherein a function of an application service interface for an application that provides a type of service translates its invocation by a function of a common service interface for that type of service to an invocation of one or more functions of the application and translates its invocation by the application to invocations of one or more functions of the common service interface.

[c39] 39. The integration server of claim 37 wherein a function of the common service interface for a type of service transforms its invocation by integration processes to invocations of one or more functions of an application service interface for an application that provides that type of service and transforms its

invocation by a function of an application service interface of the application to invocations of one or more functions of the integration processes.

[c40]            40.    The integration server of claim 37 wherein the implementations are provided by XSLT transforms.

[c41]            41.    The integration server of claim 37 wherein the common service interface and the application service interface are defined using WSDL.